

## **CLAIMS**

1. A method of allocating objects in a memory portion that includes a Young Generation and at least one Older Generation, said method comprising:

(a) determining whether at least one object should be allocated in said Young Generation in accordance with a first promotion policy exercised for promoting objects from said Young Generation to an Older Generation of said memory portion;

(b) determining a second promotion policy for said at least one object when said determining (a) determines that said object should not be allocated in said Young Generation in accordance with said first promotion policy; and

(c) allocating said at least one object in said Young Generation in accordance with said second policy when said determining (b) determines a second promotion policy for said object.

2. A method as recited in claim 2, wherein said determining (a) of whether at least one object should be allocated in accordance with a first promotion policy comprises:

determining whether said at least one object is likely to be used as temporary data.

3. A method as recited in claim 2, wherein said determining (a) of whether at least one object should be allocated in accordance with a first promotion policy comprises:

determining whether said at least one object is likely to be garbage.

4. A method as recited in claim 3, wherein said at least one object is likely to be garbage if there is at least a 50% change that said at least one object will be garbage within an acceptable time period.

5. A method as recited in claim 1, wherein said determining (a) of whether at least one object should be allocated in accordance with a first promotion policy comprises:

determining whether said at least one object is likely to be mostly garbage.

6. A method as recited in claim 5, wherein said determining of whether said at least one object is likely to be mostly garbage determines whether at least 50% of said at least one object is likely to be garbage.

7. A method as recited in claim 1,

wherein said determining (a) of whether an object should be allocated in accordance with a first promotion policy comprises:

determining whether system code is allocating said object;

and

wherein said method further comprises:

selecting a second promotion policy that postpones the promotion of said at least one object with respect to other objects allocated in accordance with said first promotion policy.

8. A method as recited in claim 1, wherein of said determining (a) of whether an object should be allocated in accordance with a first promotion policy comprises:

determining whether one or more of the following operations are being performed:

loading a class, parsing a file that represents a class, dynamic compilation, and a call to a library function that is likely to generate temporary data.

9. A method as recited in claim 8, wherein said class is a Java<sup>TM</sup> compliant class represented in a classfile, said dynamic compilations are performed in a Java<sup>TM</sup> compliant run time environment, and said library function is a Java<sup>TM</sup> compliant library method.

10. A method as recited in claim 9, wherein said Java<sup>TM</sup> compliant library method is associated with concatenation of Java<sup>TM</sup> strings.

11. A method as recited in claim 1, wherein said (c) allocating of said at least one object in said Young Generation in accordance with said second promotion policy comprises:

allocating said at least one object with a header that indicates said second policy.

12. A method as recited in claim 11, wherein said header includes a preemption indicator that indicates a garbage collection count should be preempted and said at least one object should not be promoted to said next generation.

13. A method as recited in claim 11, wherein said header includes a preemption indicator and a preemption value;

wherein said preemption indicator indicates that a garbage collection count should be preempted;

wherein said preemption value provides a preemptive garbage collection count that is used instead of a garbage collection count.

14. A method as recited in claim 11, wherein said header provides a garbage collection count that is used to determine when said at least one

object should be promoted from said Young Generation to said Older Generation.

15. A computer system, comprising:

at least one processing unit;

a memory portion that is partitioned into a Young Generation and at least one Older Generation;

a first memory allocator that operates to allocate one or more objects in said Young Generation in accordance with a first promotion policy used for promoting said one or more objects from said Young Generation to an Older Generation in said memory; and

a second memory allocator that operates to allocate one or more other objects in said Young Generation in accordance with a second promotion policy for promoting said one or more other objects from said Young Generation to an Older Generation in said memory.

16. A computer system as recited in claim 15, wherein said method further comprises:

an allocation interface that can be used to access both said first and second memory allocators, wherein said allocation interface operates to use said first or second memory allocators in accordance with an allocation selection; and

an allocation switching function that can be used to switch said allocation selection from said first memory allocator to said second memory allocator.

17. A computer system as recited in claim 15, wherein said first and second memory allocators can be directly accessed.

18. A computer system as recited in claim 15, wherein said processing unit:

determines whether at least one object should not be allocated using said first memory allocator in said Young Generation in accordance with said first promotion; and

determines said second promotion policy used by said second allocator when it is determined that at least one object should not be allocated in said Young Generation in accordance with said first promotion.

19. A computer system as recited in claim 15, wherein said second allocator allocates at least one object with a header that indicates said second promotion policy.

20. A computer system as recited in claim 15, wherein said header includes a preemption indicator indicating that a garbage collection count should be preempted and said at least one object should not be promoted to said next generation.

21. A computer system as recited in claim 20,

wherein said header includes a preemption indicator and a preemption value;

wherein said preemption indicator indicates that a garbage collection count should be preempted; and

wherein said preemption value provides a preemptive garbage collection count that is used instead of said garbage collection count.

22. A computer system as recited in claim 20, wherein said header provides a garbage collection count that is used to determine when said at least one object should be promoted from said Young Generation to said Older Generation.

23. A computer system as recited in claim 19, further comprising:  
a garbage collector that reads said header and promotes said at least one object in accordance with said header.
24. A computer system as recited in claim 23, wherein said garbage collector delays or avoids promotion of said at least one object with respect to objects allocated with said first allocator.
25. A computer system as recited in claim 15, wherein said computer system is a virtual machine.
26. A computer system as recited in claim 15, wherein said computer system is a Java<sup>TM</sup> compliant virtual machine.
27. A computer system as recited in claim 15, wherein said computer system is provided for a handheld, an embedded, or mobile device.
28. A method of garbage collecting a heap in a computing environment, said heap including a Young Generation and at least one Older Generation.  
receiving a request for allocation of one or more objects in said heap;  
determining whether promotion of said one or more objects should be delayed beyond a normal garbage collection threshold;  
generating a delay-promotion marking for said one or more objects that indicates promotion of said one or more objects should be delayed beyond said normal garbage collection threshold;  
allocating said one or more objects with said delay-promotion marking in said Young Generation of said heap;

determining whether said Young Generation should be garbage collected;

determining whether said one or more objects have reached a normal garbage collection threshold when said determining determines that said Young Generation should be garbage collected; and

not promoting said one or more objects to an Older Generation when said delay promotion marking indicates that said one or more objects should not be promoted to an Older Generation.

29. A computer readable medium including computer program code for allocating objects in a memory portion that includes a Young Generation and at least one Older Generation, said computer readable medium including comprising:

computer program code for allocating one or more objects in said Young Generation in accordance with a first promotion policy exercised for promoting objects from said Young Generation to an Older Generation of said memory portion; and

computer program code for allocating one or more other objects in said Young Generation in accordance with a second promotion policy exercised for promoting objects from said Young Generation to an Older Generation of said memory portion.

30. A computer readable medium as recited in claim 29, wherein said second promotion policy is not to promote said one or more objects from said Young Generation to said Older Generation of said memory portion.

31. A computer readable medium as recited in claim 29, wherein said second promotion policy is to delay the promotion of said one or more other objects with respect to objects allocated under said first promotion policy.